PRÁCTICA 1 ALGORÍTMICA:

Mariana Orihuela Cazorla
Adra Sánchez Ruiz
Juan Manuel Castillo Nievas
Luis Liñán Villafranca
Cristina Garrido Amador

Grupo A2

- Cálculo de la eficiencia empírica

En el caso de los algoritmos de ordenación, (burbuja, inserción, selección, mergesort, quicksort y heapsort) para medir el tiempo empleado por el algoritmo para la ordenación de un vector según su tamaño, el tamaño viene dado por el número de componentes del vector a ordenar.

Sin embargo, para el algoritmo de Floyd que calcula los mínimos entere todos los padres de nodos en un grafo dirigido, el tamaño es el número de nodos del grafo.

Por otro lado, para el algoritmo de Fibonacci lo que calculamos la sucesión de Fibonacci para el número que damos de entrada.

En todos ellos, el cálculo del tiempo lo hemos realizado mediante el uso de la biblioteca chrono de la STL, siguiendo los pasos especificados en el pdf para medir el valor del reloj antes y después de la ejecución del código y compilando añadiendo a la orden -std=c++11.

Así, vamos probando con diferentes tamaños de entrada con la ayuda de una macro y se almacenan en el correspondiente fichero con la extensión .dat.

A continuación con ayuda del gnuplot hemos creado las correspondientes gráficas con sus correspondientes etiquetas. Para hacer un buen ajuste hemos utilizado las siguientes funciones:

- Para eficiencia $O(n^2)$ la función es → f(x) = a0*x*x+a1*x+a2
- Para eficiencia O(nlogn) la funcion es → f(x) = a0*x*log(x)
- Para el algoritmo de Fibonacci la función es → f(x) = a0**x
- Para el algoritmo de Floyd la función es → f(x) = a0*x*x*x+a1*x*x+a2*x+a3

A continuación se encuentran las tablas de los valores de los algoritmos para distinto orden de eficiencia.
En el caso de los 6 primeros algoritmos, el tamaño de entrada va variando de 1000 en 1000, para el algoritmo de Fibonacci es de 2 en 2 y por último, para el algoritmo de Floyd de 1500 en 1500.

■ Orden n^2

Vector size	Burbuja	Selección	Inserción
1000	0.00249057	0.00137404	0.00129747
2000	0.0106799	0.00507993	0.00503552
3000	0.0178483	0.0114571	0.0125135
4000	0.0325178	0.0202148	0.0219809
5000	0.0545374	0.0306429	0.0329178
6000	0.0791116	0.0454891	0.0481654
7000	0.117478	0.0597942	0.0640392
8000	0.157062	0.0781305	0.0841856
9000	0.204648	0.0984296	0.103707
10000	0.244086	0.125935	0.1307
11000	0.307099	0.150902	0.155586
12000	0.366801	0.180957	0.183808
13000	0.436182	0.213751	0.216433
14000	0.515567	0.24248	0.254292
15000	0.604836	0.276531	0.286249
16000	0.691392	0.33454	0.332186
17000	0.78145	0.360354	0.370643
18000	0.894802	0.404458	0.412793
19000	0.991668	0.44766	0.461129
20000	1.09821	0.49328	0.510552
21000	1.23811	0.540309	0.565047
22000	1.33587	0.5965	0.636048
23000	1.46577	0.654383	0.696059
24000	1.58152	0.706164	0.743055
25000	1.73501	0.765662	0.821587

■ Orden O(nlogn)

Vector size	Quicksort	Mergesort	Heapsort
1000	0.000112235	0.000171407	0.000149853
2000	0.000248455	0.000390416	0.000520264
3000	0.000342316	0.000685591	0.000838432
4000	0.000453731	0.000757152	0.000952279
5000	0.000593879	0.001079	0.000872722
6000	0.000622042	0.00110847	0.000978596
7000	0.000781037	0.00117944	0.000961212
8000	0.000812354	0.0013048	0.00110655
9000	0.000988758	0.00154032	0.00131779
10000	0.00111115	0.00185227	0.001641
11000	0.00134213	0.00206365	0.00158432
12000	0.00138318	0.00246043	0.00174526
13000	0.00145887	0.00203945	0.00194725
14000	0.00155896	0.00237961	0.0021037
15000	0.00165263	0.00248817	0.00229874
16000	0.00193017	0.00268028	0.00250462
17000	0.00197597	0.00294842	0.00262725
18000	0.00195717	0.0032173	0.00281652
19000	0.00214444	0.00351858	0.00293355
20000	0.00229063	0.0037565	0.00314785
21000	0.0024023	0.00399967	0.00323868
22000	0.00261219	0.00428035	0.00339132
23000	0.00258809	0.00448928	0.00359764
24000	0.00264344	0.00477767	0.00370761
25000	0.00296003	0.00505949	0.00386291

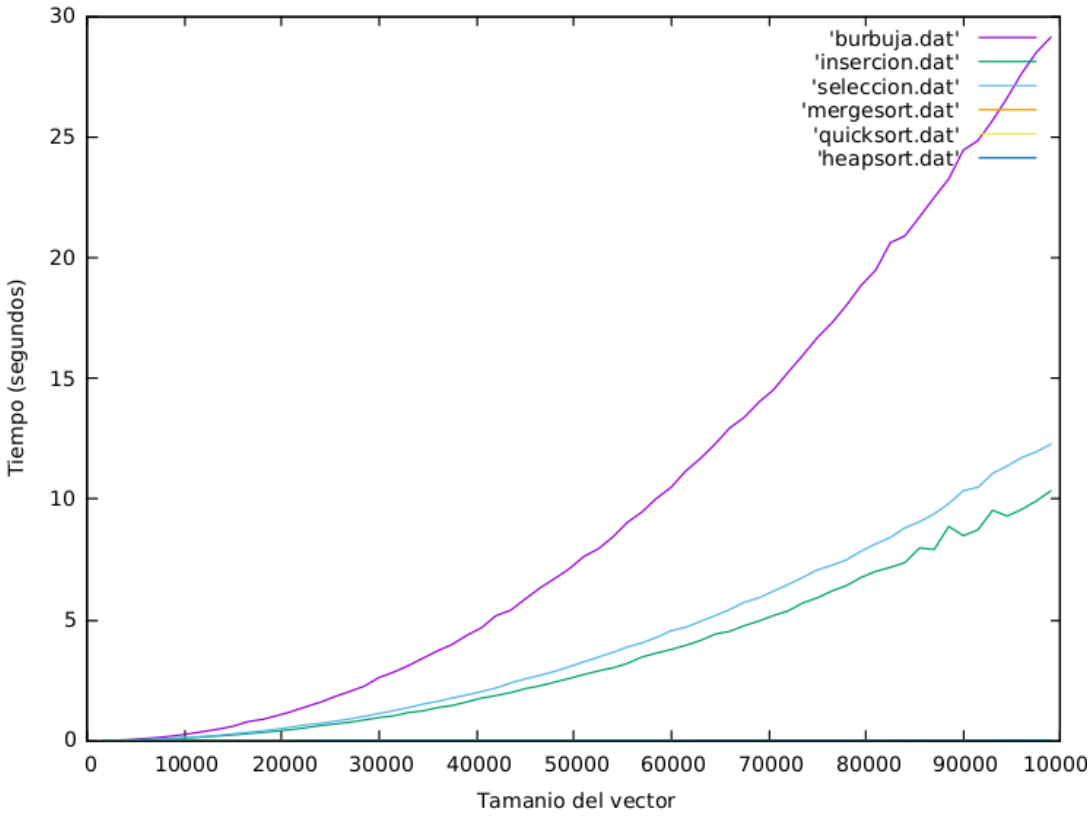
■ Orden fibonacci

Vector size	Fibonacci
2	1,18E-07
4	1,72E-07
6	2,60E-07
8	5,44E-07
10	8,74E-07
12	1,49E-06
14	3,25E-06
16	1,10E-05
18	1,89E-05
20	4,82E-05
22	0,000125474
24	0,000324169
26	0,000847802
28	0,00223019
30	0,00602786
32	0,0149237
34	0,0388415
36	0,103872
38	0,267924
40	0,709455
42	1,84618
44	4,73836
46	12,2998
48	31,9394
50	83,601
52	218,503

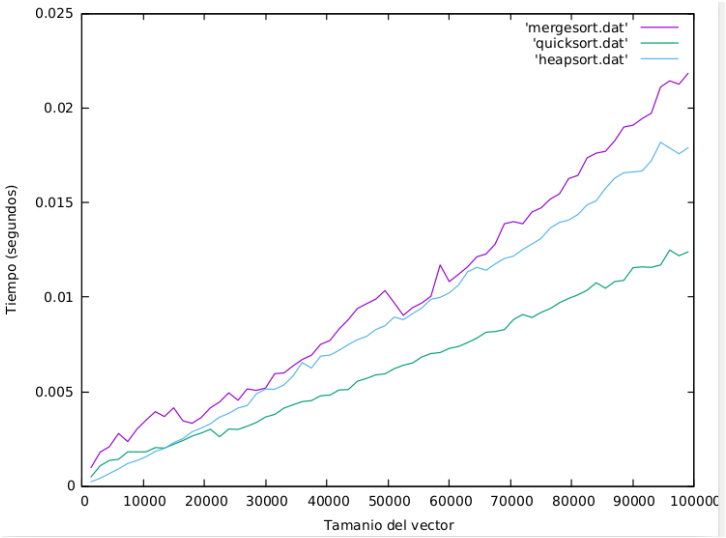
■ Orden O(n^3)

Vector size	Floyd
1500	0.0222007
3000	0.0876984
4500	0.197071
6000	0.352675
7500	0.548417
9000	0.800994
10500	1.0998
12000	1.40472
13500	1.81587
15000	2.19983
16500	2.66324
18000	3.15866
19500	3.83632
21000	4.39773
22500	4.98494
24000	5.67746
25500	6.3687
27000	7.39284
28500	8.05695
30000	8.87498
31500	9.77879
33000	10.6156
34500	11.7399
36000	13.0556
37500	13.8161

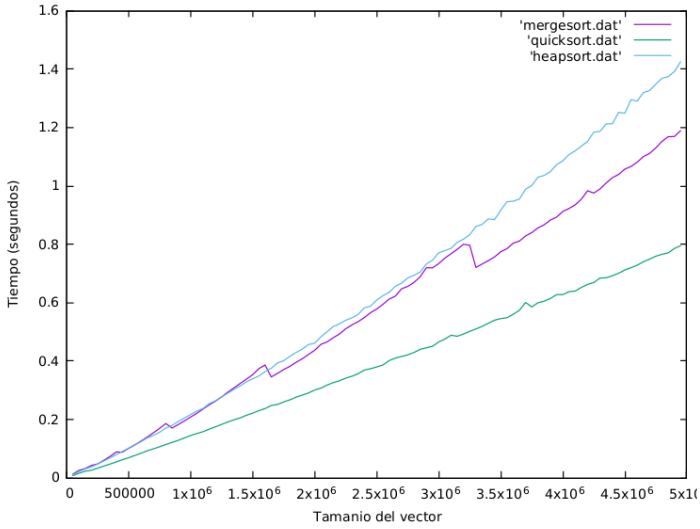
- Gráficos para las tablas de los algoritmos.
  - Gráfica para comparar los algoritmos de ordenación:



Ya que en esta gráfica no se aprecian las líneas correspondientes a mergesort, quicksort, y heapsort por que sus tiempos no son superiores a 1 segundo, a continuación se muestran por separado respecto de los de burbuja, inserción y selección desde dos computadores distintos:



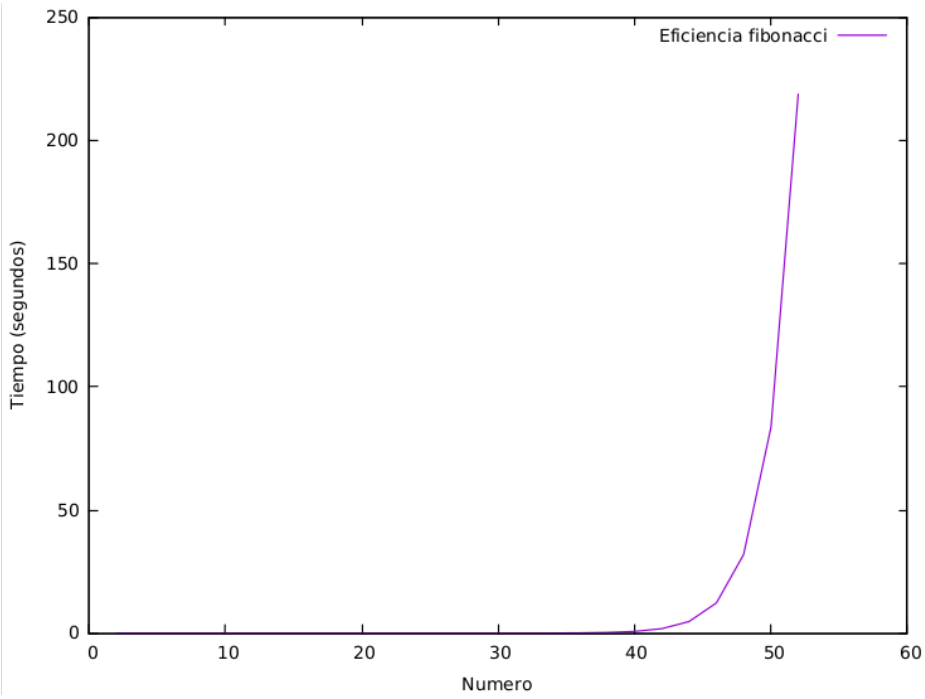

```
Arquitectura:          x86_64
modo(s) de operación de las CPUs:32-bit, 64-bit
Orden de bytes:        Little Endian
CPU(s):                8
On-line CPU(s) list:   0-7
Hilo(s) de procesamiento por núcleo:2
Núcleo(s) por «socket»:4
Socket(s):             1
Modo(s) NUMA:          1
ID de fabricante:      GenuineIntel
Familia de CPU:        6
Modelo:                60
Model name:            Intel(R) Core(TM) i7-4700H
CPU @ 2.40GHz
Revisión:              3
CPU MHz:               1258.687
CPU max MHz:           3400,0000
CPU min MHz:           800,0000
BogoMIPS:              4789.01
Virtualización:        VT-x
Caché L1d:             32K
Caché L1i:             32K
Caché L2:              256K
Caché L3:              6144K
NUMA node0 CPU(s):     0-7
```



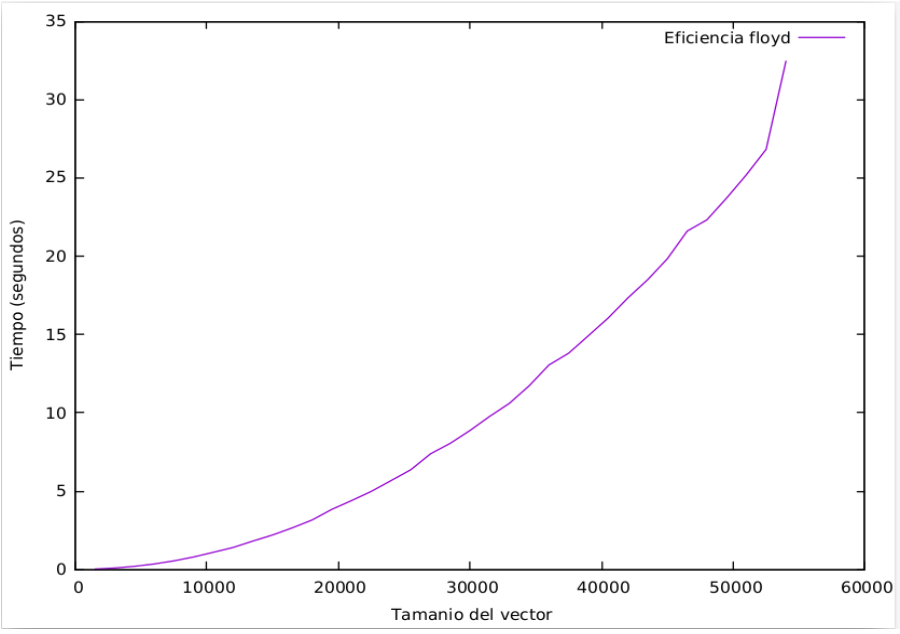

```
Arquitectura:          x86_64
modo(s) de operación de las CPUs:32-bit, 64-bit
Orden de bytes:        Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Hilo(s) de procesamiento por núcleo:2
Núcleo(s) por «socket»:2
Socket(s):             1
Modo(s) NUMA:          1
ID de fabricante:      GenuineIntel
Familia de CPU:        6
Modelo:                61
Revisión:              4
CPU MHz:               2199.914
BogoMIPS:              4393.58
Virtualización:        VT-x
Caché L1d:             32K
Caché L1i:             32K
Caché L2:              256K
Caché L3:              3072K
NUMA node0 CPU(s):     0-3
```

Para los otros 2 algoritmos, se adjuntan a continuación sus gráficas correspondientes:

- Fibonacci



- Floyd



- Cálculo de la eficiencia híbrida

Para poder aplicar un algoritmo en una situación concreta conociendo de la forma más exacta posible la ecuación del tiempo del mismo, es necesario conocer el valor de las constantes ocultas mediante la realización de un ajuste de regresión por mínimos cuadrados entre la función y los puntos de la gráfica obtenidos al calcular la eficiencia teórica.
Así, conoceremos el tiempo aproximado que tardará cada algoritmo para cualquier entrada de tamaño n.

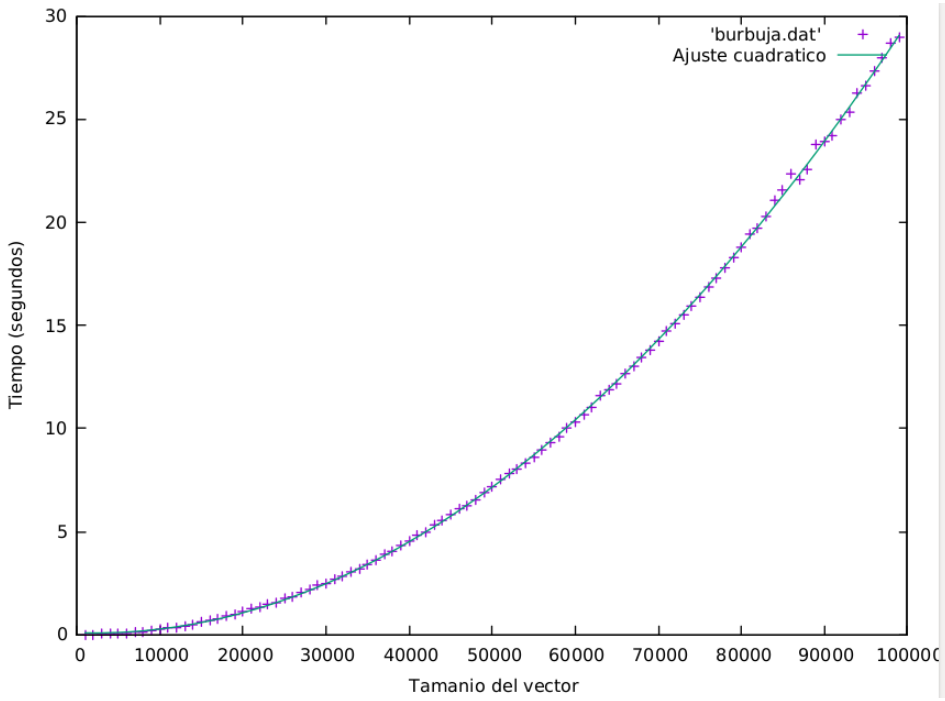
Para realizar lo anterior utilizamos gnuplot con la orden fit de modo que además de obtener los valores de las constantes bajo la sección Final set of parameters, una gráfica de la curva ajustada.

Las gráficas híbridas de los diferentes algoritmos se muestran a continuación. La primera gráfica se corresponde al ajuste bueno y debajo indicamos lo que hemos obtenido al ajustarla. La segunda gráfica es un ajuste malo.

```
Arquitectura:          x86_64
modo(s) de operación de las CPUs:32-bit, 64-bit
Orden de bytes:        Little Endian
CPU(s):                8
On-line CPU(s) list:   0-7
Hilo(s) de procesamiento por núcleo:2
Núcleo(s) por «socket»:4
Socket(s):             1
Modo(s) NUMA:          1
ID de fabricante:      GenuineIntel
Familia de CPU:        6
Modelo:                60
Model name:            Intel(R) Core(TM) i7-4700H
CPU @ 2.40GHz
Revisión:              3
CPU MHz:               1258.687
CPU max MHz:           3400,0000
CPU min MHz:           800,0000
BogoMIPS:              4789.01
Virtualización:        VT-x
Caché L1d:             32K
Caché L1i:             32K
Caché L2:              256K
Caché L3:              6144K
NUMA node0 CPU(s):     0-7
```

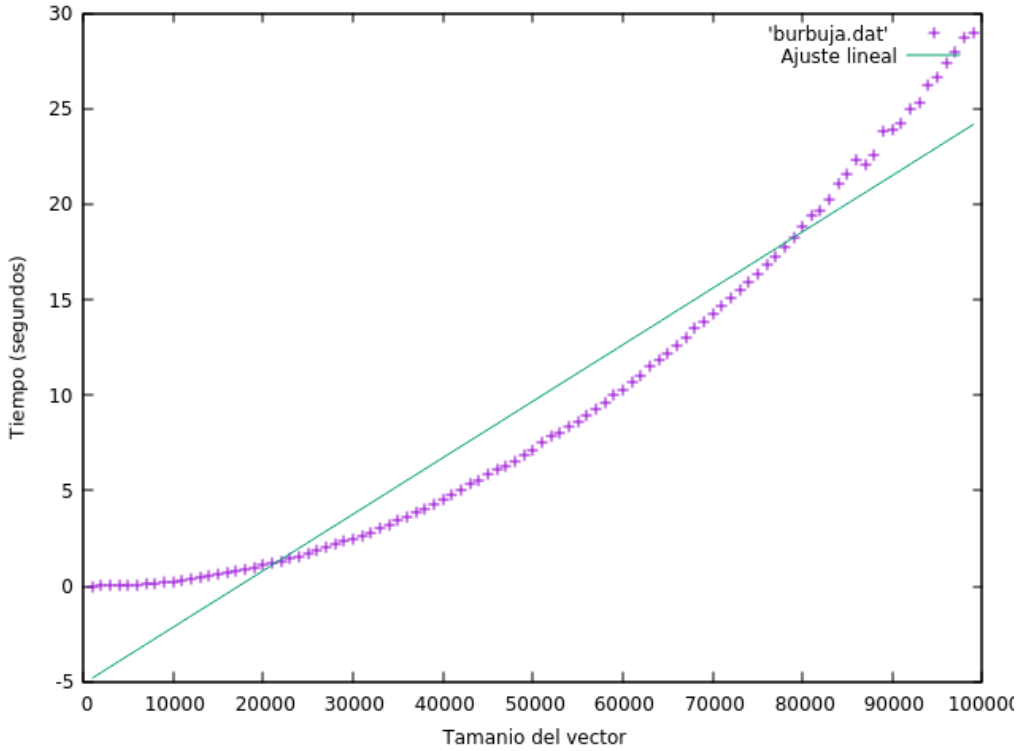
Los resultados siguientes se han realizado con un ordenador de estas características

- Burbuja

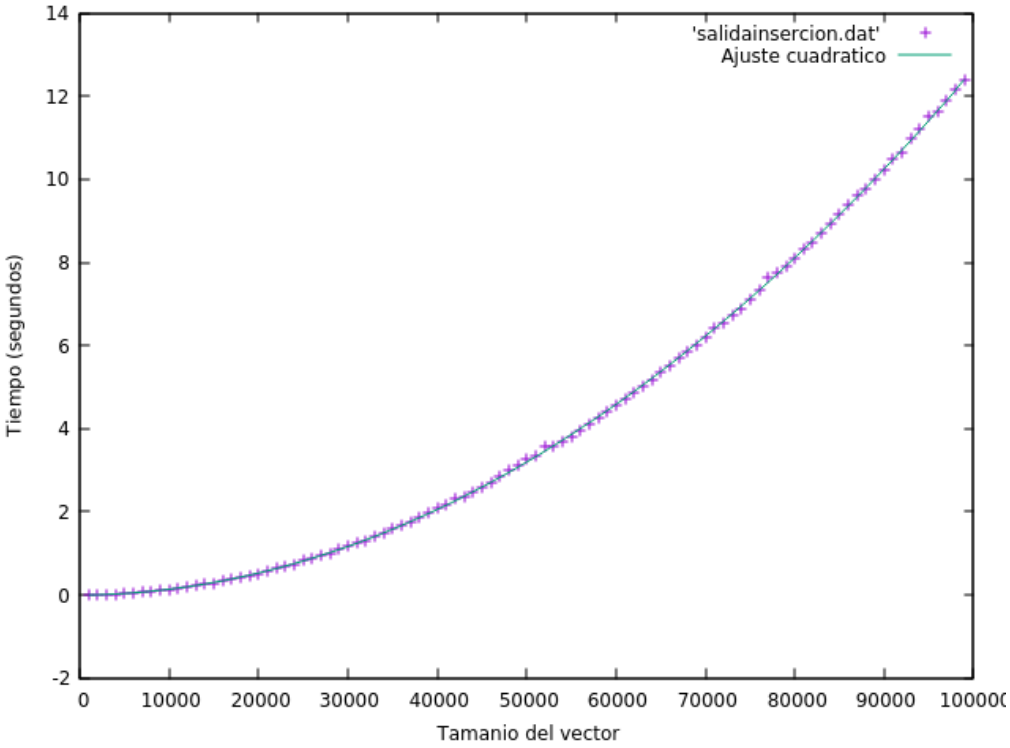


f(x) = a0*x*x+a1*x+a2

	Final set of parameters	Asymptotic Standard Error	
a0	= 3.09267e-09	+/- 1.593e-11	(0.5151%)
a1	= -1.35208e-05	+/- 1.644e-06	(12.16%)
a2	= 0.0941519	+/- 0.03562	(37.83%)

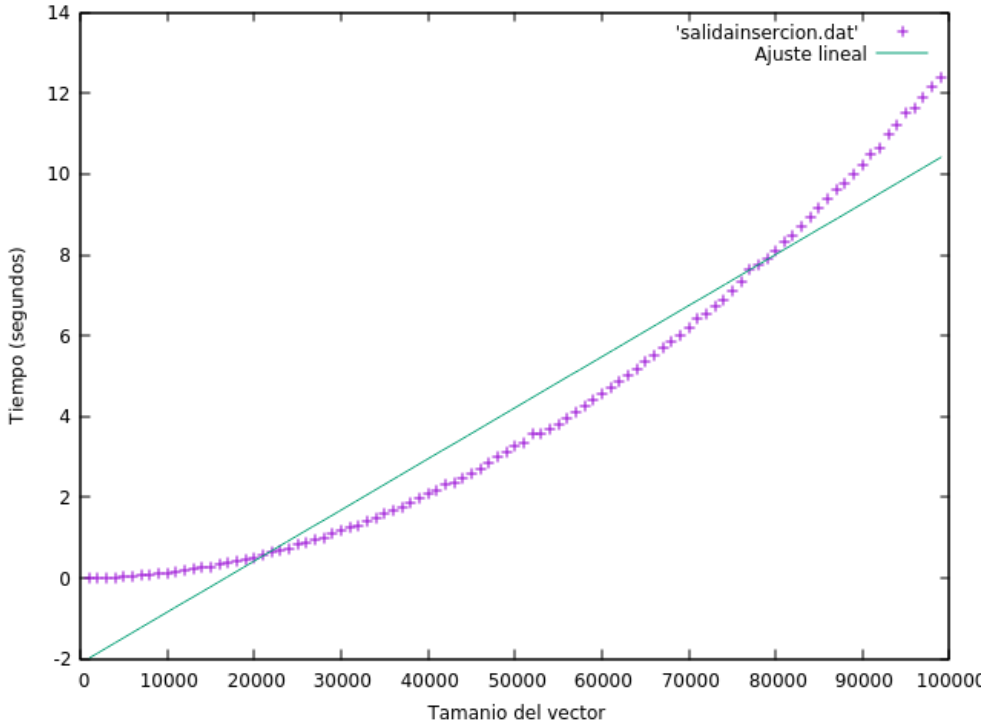


- Inserción

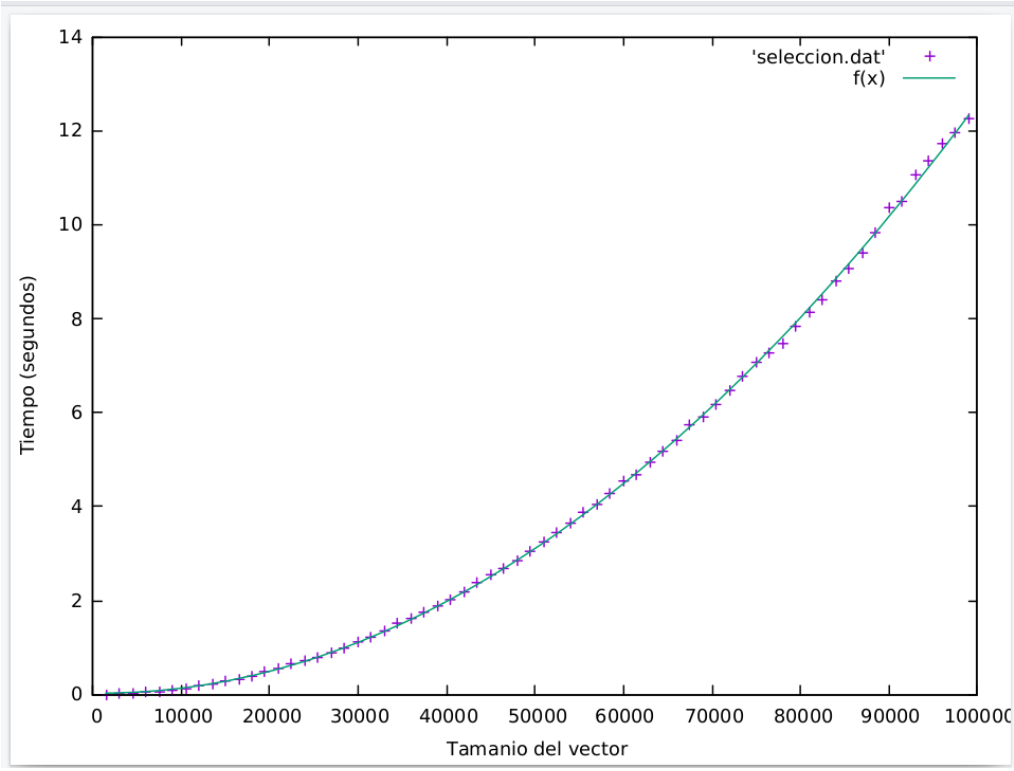


f(x) = a0*x*x+a1*x+a2

Final set of parameters		Asymptotic Standard Error	
==========		==========	
a0	= 1.24814e-09	+/- 4.247e-12	(0.3403%)
a1	= 1.59282e-06	+/- 4.384e-07	(27.52%)
a2	= -0.00532981	+/- 0.009497	(178.2%)

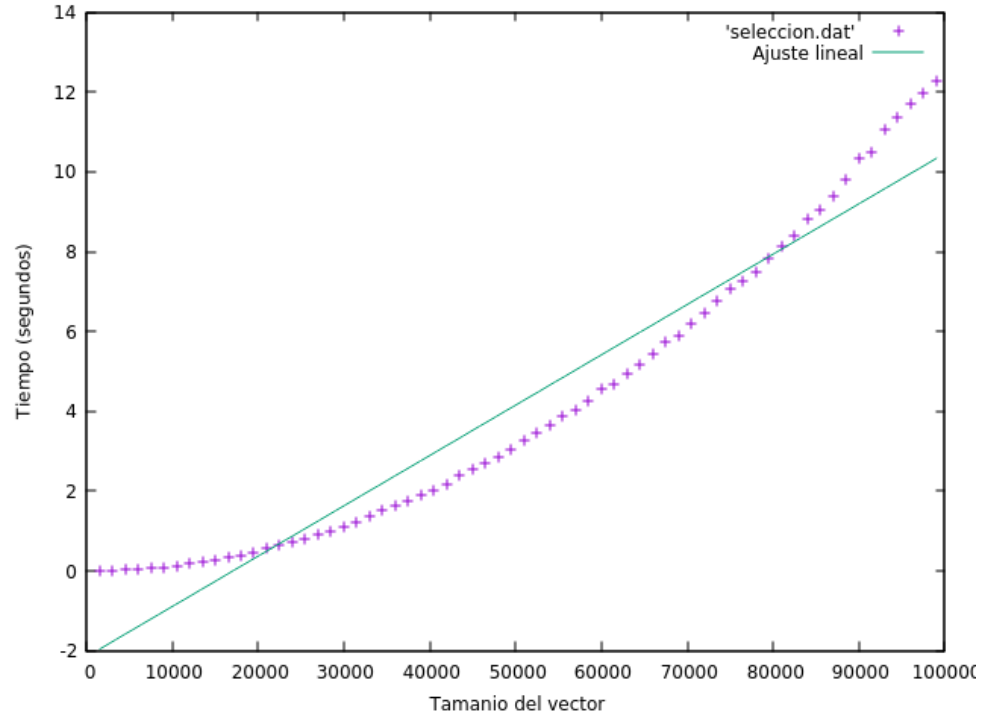


- Selección:

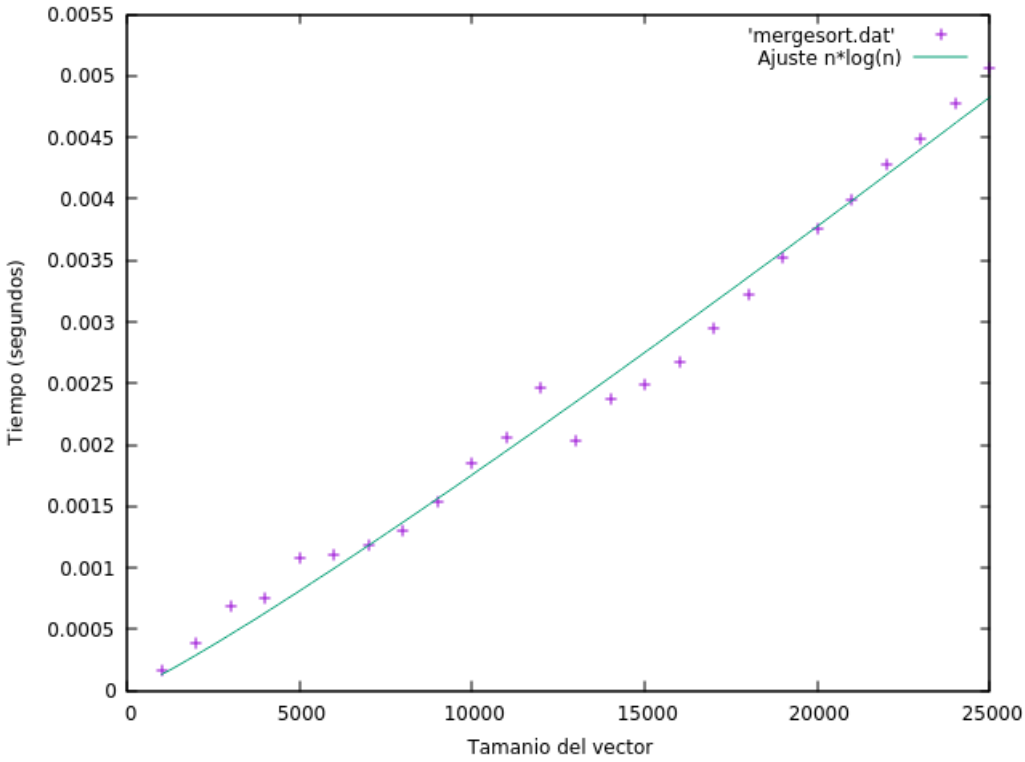


f(x) = a0*x*x+a1*x+a2

Final set of parameters		Asymptotic Standard Error	
a0	= 1.27856e-09	+/- 9.882e-12	(0.7729%)
a1	= -2.37372e-06	+/- 1.025e-06	(43.17%)
a2	= 0.0312793	+/- 0.02232	(71.35%)

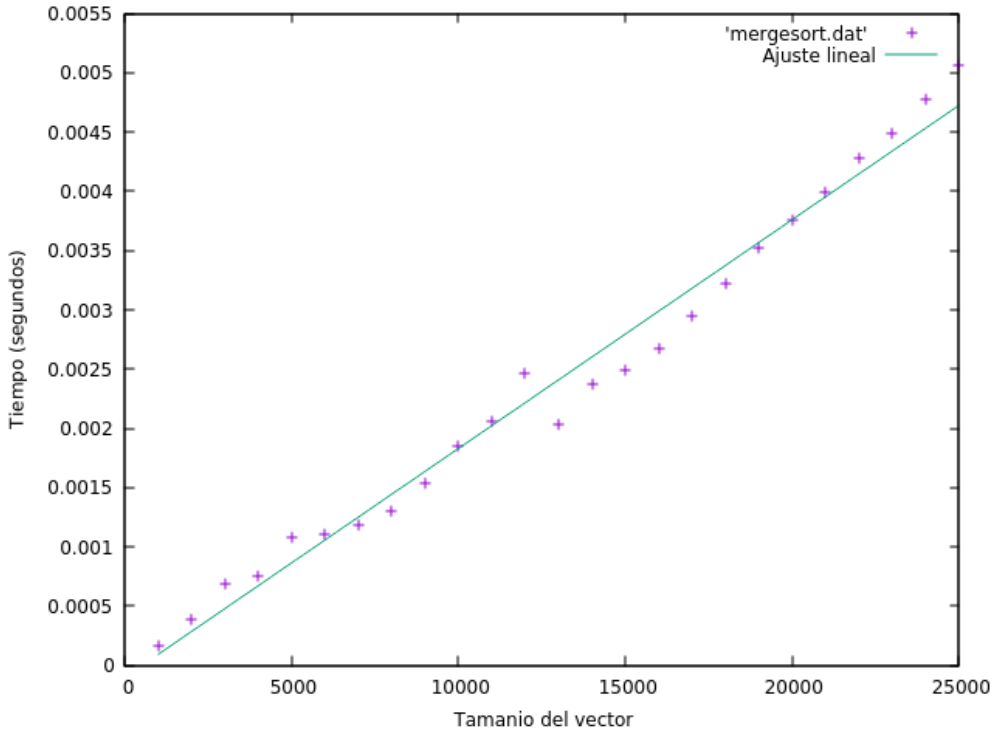


- Mergesort

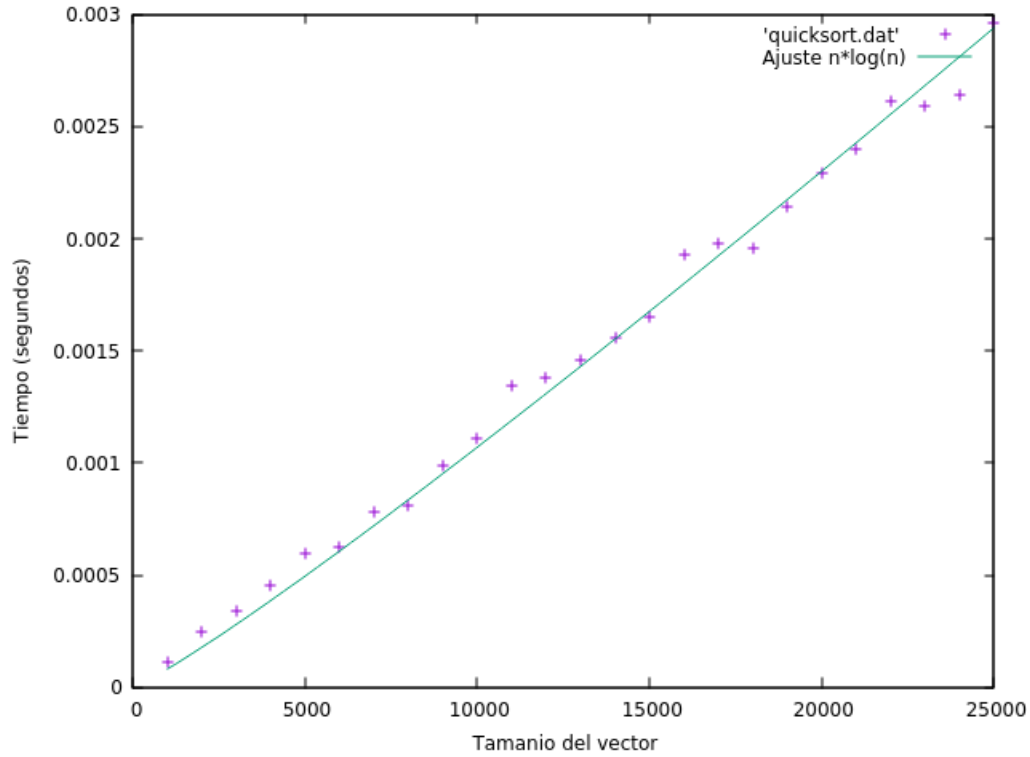


f(x) = a0*x*log(x)

Final set of parameters		Asymptotic Standard Error	
==========		==========	
a0	= 1.90722e-08	+/- 2.368e-10	(1.241%)

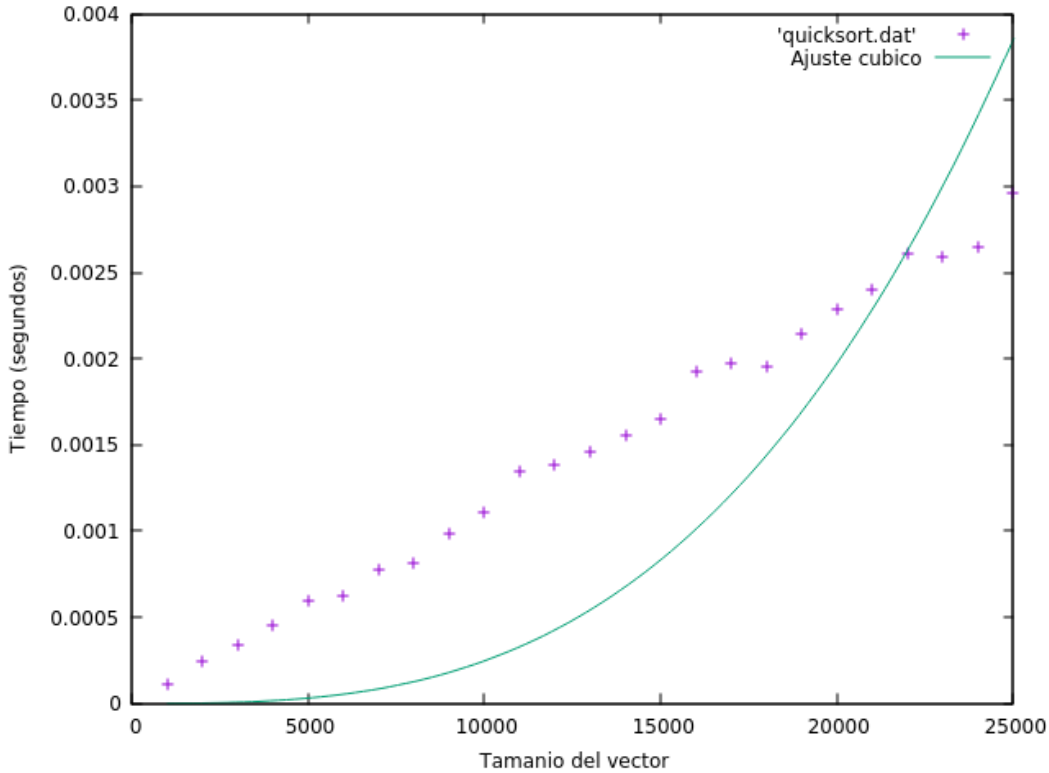


- Quicksort

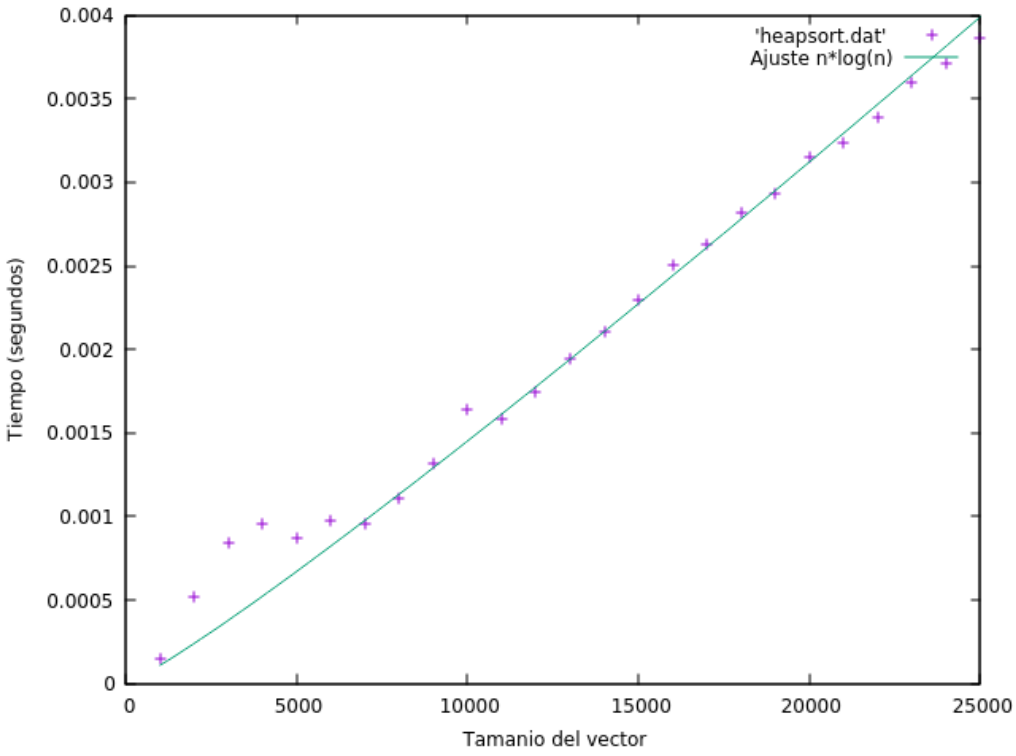


f(x) = a0*x*log(x)

Final set of parameters		Asymptotic Standard Error	
a0	= 1.16137e-08	+/- 1.022e-10	(0.8799%)

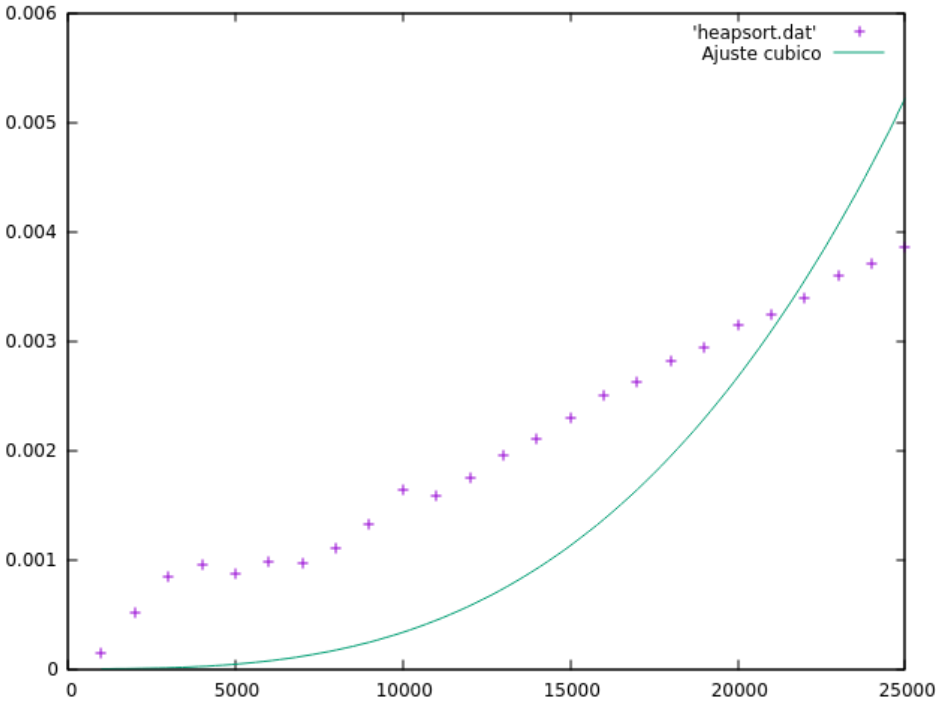


- Heapsort

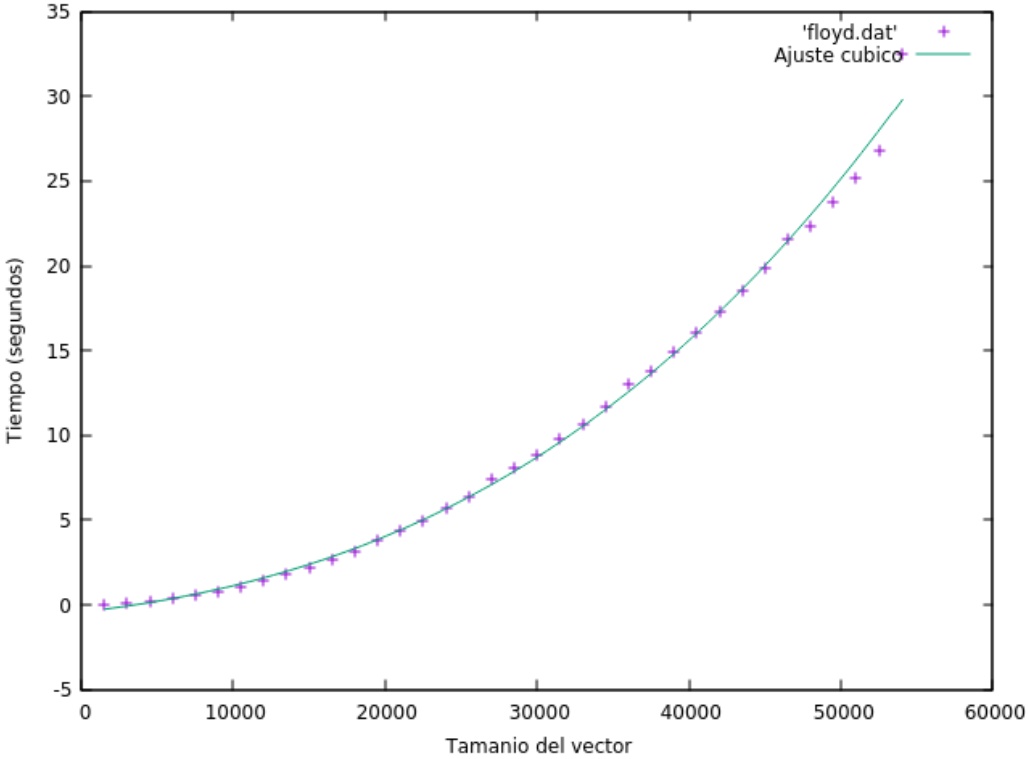


f(x) = a0*x*log(x)

Final set of parameters		Asymptotic Standard Error	
a0	= 1.57591e-08	+/- 5.579e-11	(0.354%)

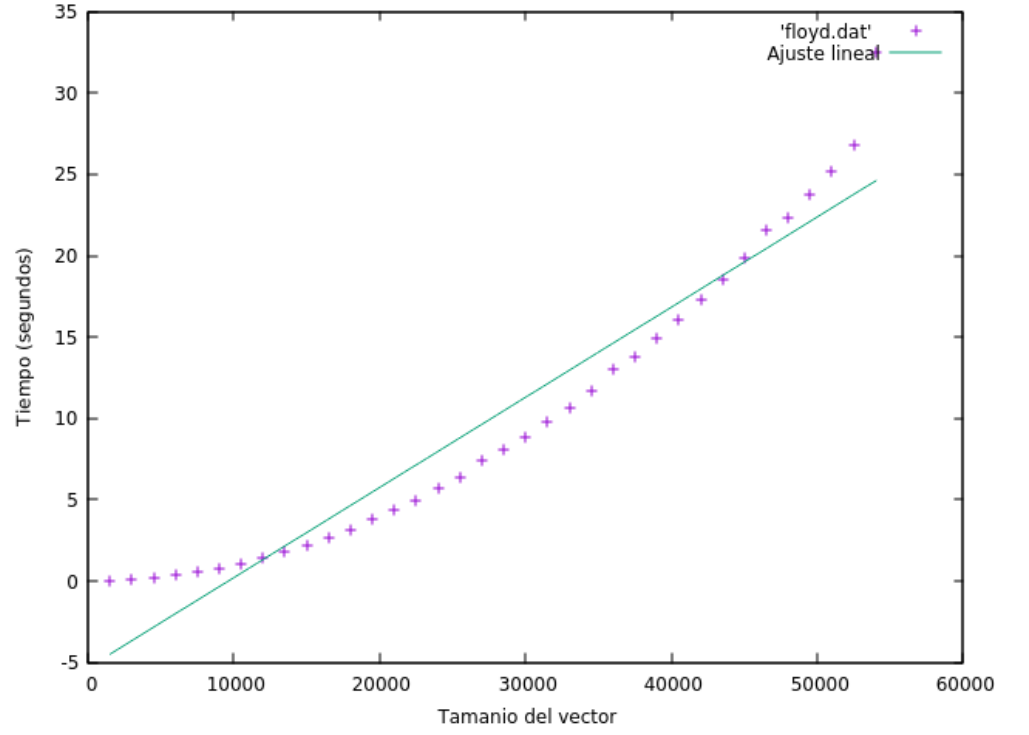


- Floyd

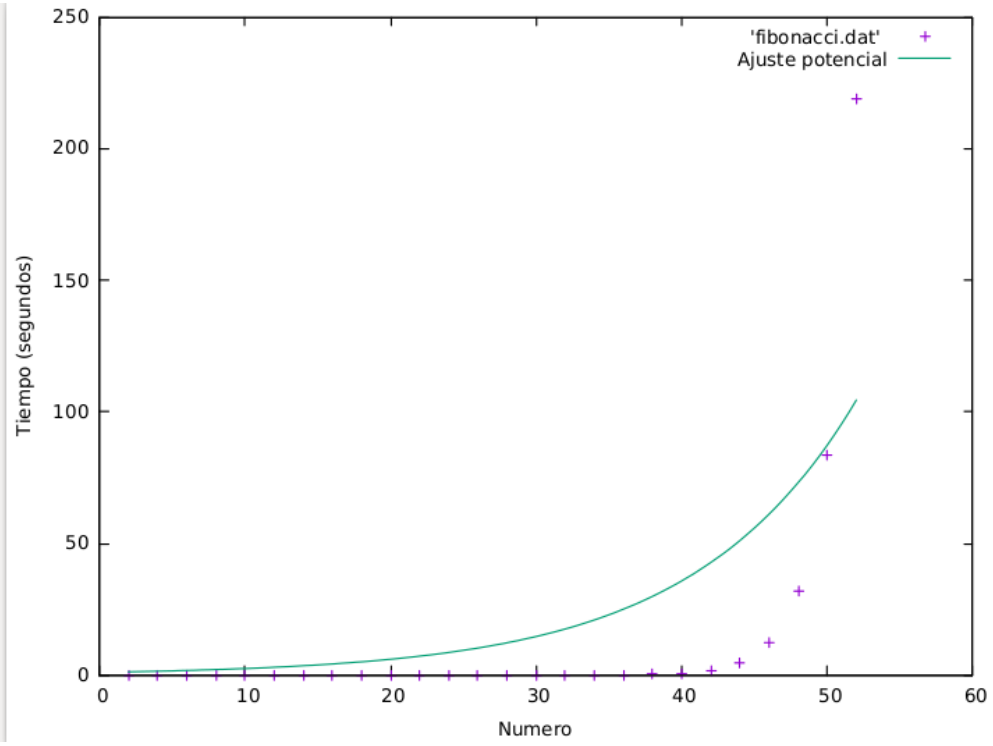


f(x) = a0*x*x*x+a1*x*x+a2*x+a3

Final set of parameters		Asymptotic Standard Error	
a0	= 7.19945e-14	+/- 3.404e-14	(47.28%)
a1	= 4.56758e-09	+/- 2.872e-09	(62.87%)
a2	= 0.000102822	+/- 6.91e-05	(67.2%)
a3	= -0.42418	+/- 0.4489	(105.8%)

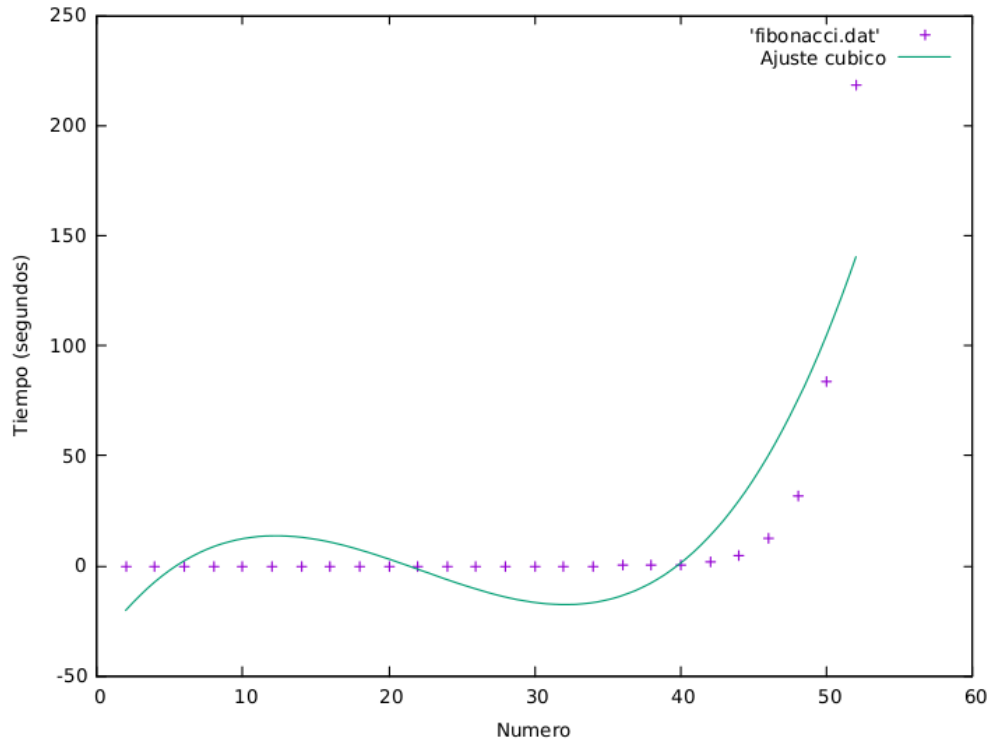


- Fibonacci



f(x) = a0**x

	Final set of parameters		Asymptotic Standard Error	
==========		==========		
a0	= 1.09349		+/- 0.003728	(0.3409%)

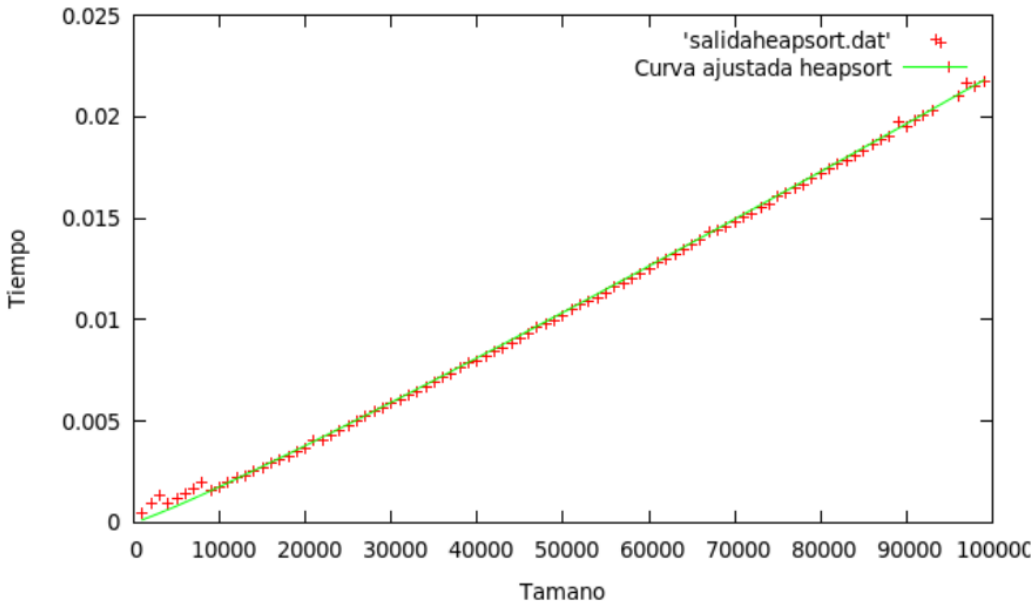


- Comparando con otros ordenadores

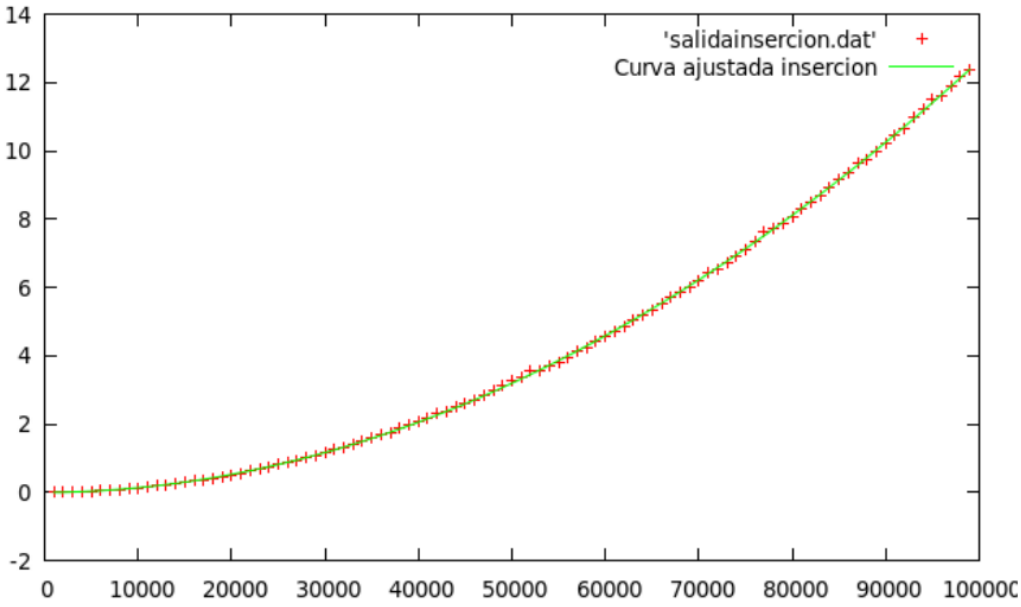
A continuación vamos a mostrar las gráficas que hemos obtenido con los diferentes algoritmos en diferentes ordenadores. La eficiencia empírica se muestra en los puntos y la eficiencia híbrida en líneas, para poder ver cómo en diferentes ordenadores también varía el ajuste.

```
Arquitectura:          x86_64
modo(s) de operación de las CPUs:32-bit, 64-bit
Orden de bytes:        Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Hilo(s) de procesamiento por núcleo:2
Núcleo(s) por «socket»:2
Socket(s):             1
Modo(s) NUMA:          1
ID de fabricante:      GenuineIntel
Familia de CPU:        6
Modelo:                61
Revisión:              4
CPU MHz:               2199.914
BogoMIPS:              4393.58
Virtualización:        VT-x
Caché L1d:             32K
Caché L1i:             32K
Caché L2:              256K
Caché L3:              3072K
NUMA node0 CPU(s):     0-3
```

A continuación se muestran los algoritmos de heapsort e inserción con las características de este ordenador



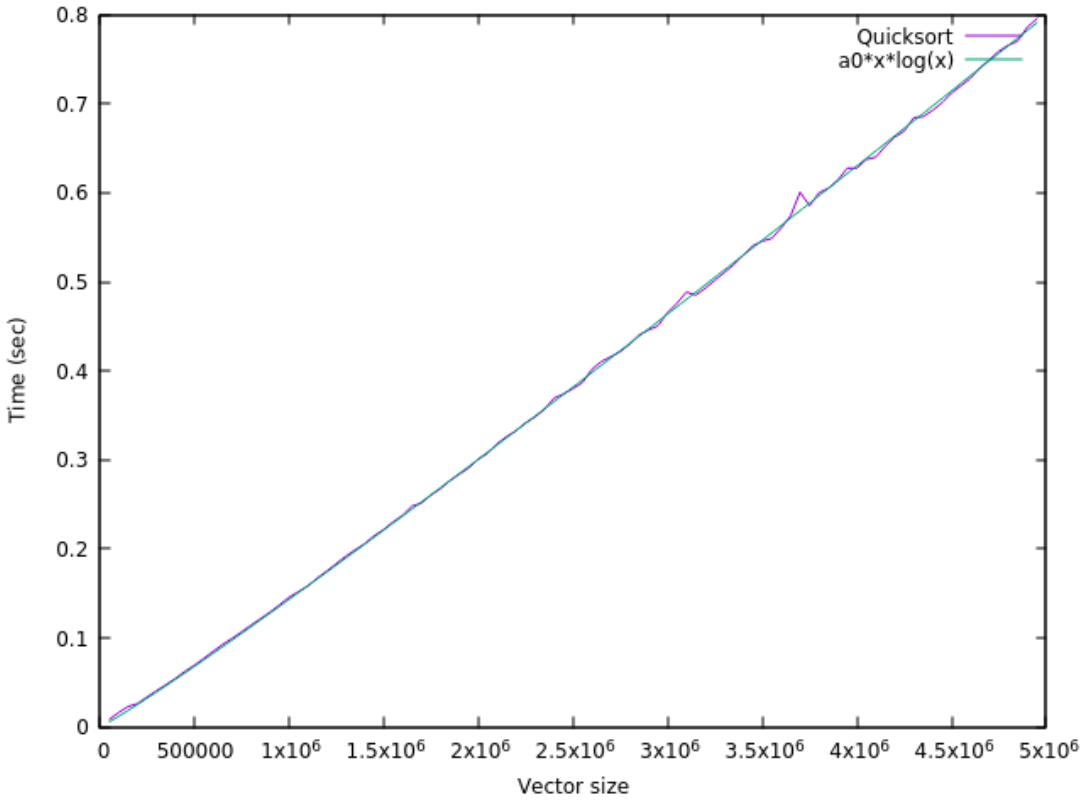
Vector size	Heapsort
1000	0.000438058
2000	0.000953388
3000	0.00133457
4000	0.000922218
5000	0.00118435
6000	0.00145488
7000	0.00166276
8000	0.00199412
9000	0.00158855
10000	0.00176384
11000	0.00196756
12000	0.00217505
13000	0.00229628
14000	0.00251431
15000	0.00270857
16000	0.00291928
17000	0.00308716
18000	0.00326992
19000	0.00351063
20000	0.00368085
21000	0.00408452
22000	0.00406872
23000	0.00432625
24000	0.00450405
25000	0.00474299



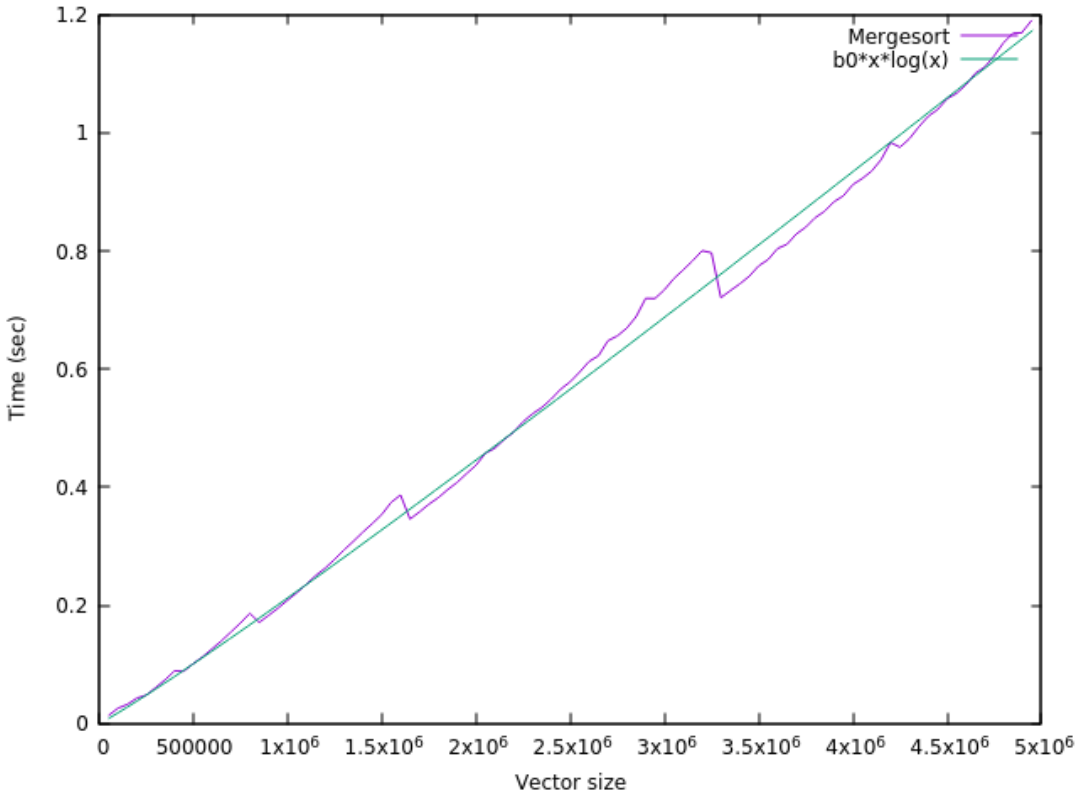
Vector size	Inserción
1000	0.00129747
2000	0.00503552
3000	0.0125135
4000	0.0219809
5000	0.0329178
6000	0.0481654
7000	0.0640392
8000	0.0841856
9000	0.103707
10000	0.1307
11000	0.155586
12000	0.183808
13000	0.216433
14000	0.254292
15000	0.286249
16000	0.332186
17000	0.370643
18000	0.412793
19000	0.461129
20000	0.510552
21000	0.565047
22000	0.636048
23000	0.696059
24000	0.743055
25000	0.821587

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                8
On-line CPU(s) list:   0-7
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 94
Model name:            Intel(R) Core(TM) i7-6700HQ
CPU @ 2.60GHz
Stepping:              3
CPU MHz:               899.938
CPU max MHz:           3500.0000
CPU min MHz:           800.0000
BogoMIPS:              5186.00
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              6144K
NUMA node0 CPU(s):     0-7
```

A continuación se muestran los algoritmos de quicksort, mergesort y heapsort con las características de este ordenador, y la comparación de estos tres.

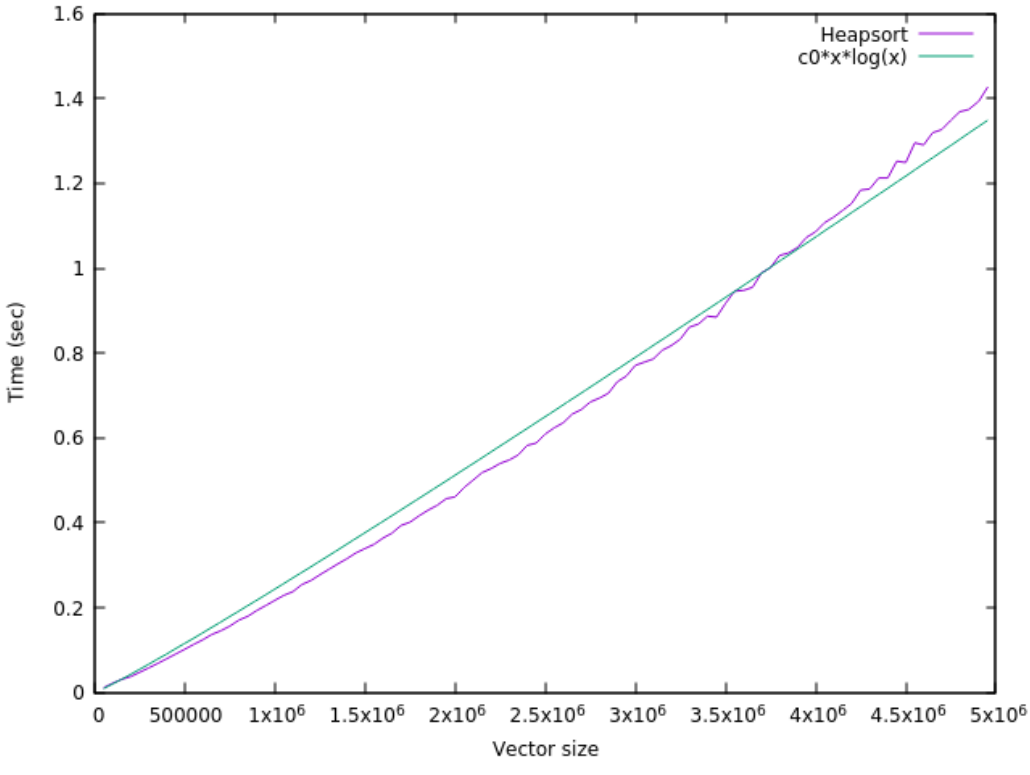


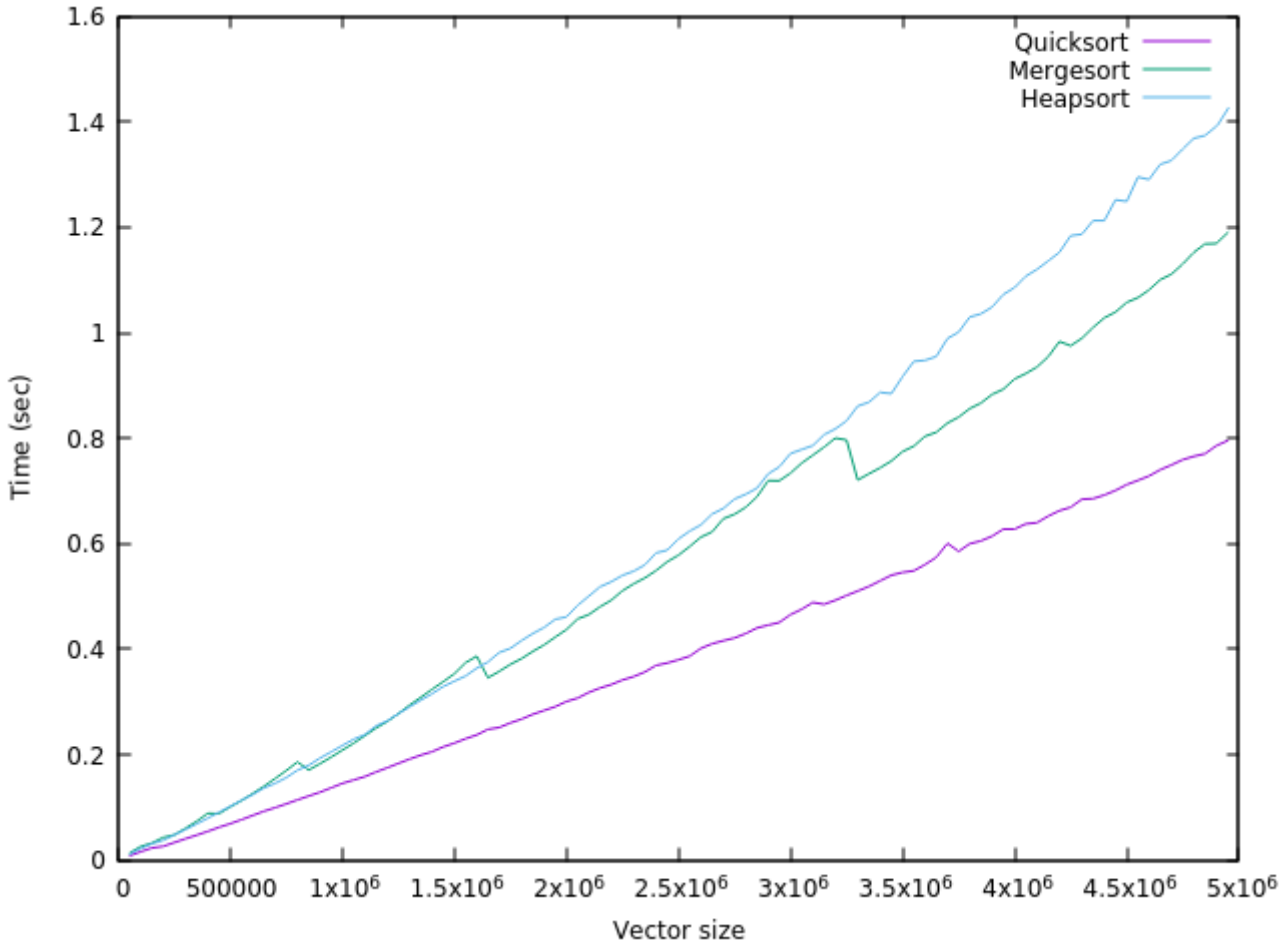
Vector size	Quicksort
50000	0.00841
100000	0.016144
150000	0.022992
200000	0.026274
250000	0.033615
300000	0.040666
350000	0.047574
400000	0.054568
450000	0.062299
500000	0.069066
550000	0.076929
600000	0.084911
650000	0.092664
700000	0.099492
…	…
4200000	0.662701
4250000	0.669171
4300000	0.684369
4350000	0.685354
4400000	0.692415
4450000	0.700967
4500000	0.712053
4550000	0.720128
4600000	0.728433
4650000	0.740069
4700000	0.749305
4750000	0.759011
4800000	0.765692
4850000	0.770436
4900000	0.785937
4950000	0.795342



Vector size	Mergesort
50000	0.013294
100000	0.025783
150000	0.032272
200000	0.042872
250000	0.04782
300000	0.060559
350000	0.073807
400000	0.089043
450000	0.088283
500000	0.101158
550000	0.112963
600000	0.126132
650000	0.139892
700000	0.154778
…	…
4200000	0.98349
4250000	0.975343
4300000	0.989572
4350000	1.00986
4400000	1.02798
4450000	1.03936
4500000	1.05761
4550000	1.06654
4600000	1.08098
4650000	1.1005
4700000	1.11117
4750000	1.12997
4800000	1.15249
4850000	1.16827
4900000	1.16928
4950000	1.18918

Vector size	Heapsort
50000	0.011243
100000	0.021899
150000	0.0313
200000	0.037516
250000	0.047913
300000	0.058058
350000	0.068704
400000	0.079636
450000	0.090904
500000	0.101741
550000	0.113638
600000	0.124404
650000	0.136903
700000	0.145558
…	…
4200000	1.15239
4250000	1.18388
4300000	1.18649
4350000	1.21232
4400000	1.21259
4450000	1.25182
4500000	1.249
4550000	1.29528
4600000	1.29025
4650000	1.31923
4700000	1.32655
4750000	1.34807
4800000	1.36851
4850000	1.37397
4900000	1.39142
4950000	1.42512




Quicksort
Mergesort
Heapsort
Time (sec)
Vector size
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
0
500000
1x10^6
1.5x10^6
2x10^6
2.5x10^6
3x10^6
3.5x10^6
4x10^6
4.5x10^6
5x10^6